This listing of claims will replace all prior versions, and listings, of claims in the application:

## LISTING OF CLAIMS

1. (Currently Amended) A method of generating a document, the method comprising:

creating a current transaction data set;

establishing at least one computer-processable dynamic document structure;

establishing a set of computer-processable rules in accordance with a rules markup language;

configuring each rule in the set of computer-processable rules to be embedded in the at least one ~~or more~~ computer-processable dynamic document structure[[s]] and to determine [[the]] content to be included in at least one instance of a document generated from the at least one ~~of the one or more~~ computer-processable dynamic document structure[[s]] when each rule is executed based on the current transaction data set;

creating a computer-implemented database;

storing each rule in the set of computer-processable rules in the database;

storing content in the database;

associating the stored content with one or more rules from the set of computer-processable rules in the database;

configuring ~~each~~ the at least one computer-processable dynamic document structure to have a tree-architecture, to resolve to the at least ~~one or more~~ instance[[s]] of a document, and to include document content embedded with ~~including~~ one or more rules from the set of computer-processable ~~embedded~~ rules; and

resolving, with a computer processor and in accordance with the current transaction data set, the at least one dynamic document structure by executing the one or more ~~embedded~~ rules from the set of computer-processable rules embedded in the document content based on the current transaction data set to create a specific instance of a document in a static form.

2. (Previously Presented) A method as claimed in claim 1, wherein establishing a set of computer-processable rules in accordance with a rules markup language includes creating a schema having a conditions element.

3. (Previously Presented) A method as claimed in claim 1, wherein establishing a set of computer-processable rules in accordance with a rules markup language includes creating a schema having a choose element.

4. (Previously Presented) A method as claimed in claim 1, wherein establishing a set of computer-processable rules in accordance with a rules markup language includes creating a schema having an iterators element.

5. (Previously Presented) A method as claimed in claim 1, wherein establishing a set of computer-processable rules in accordance with a rules markup language includes creating a schema having a functions element.

6. (Previously Presented) A method as claimed in claim 1, wherein establishing a set of computer-processable rules in accordance with a rules markup language includes creating a schema having a conditions element, a choose element, an iterators element, and a functions element.

7. (Previously Presented) A method as claimed in claim 1, wherein establishing a set of computer-processable rules in accordance with a rules markup language includes creating a schema having an external interface element that is configured to be resolved into a value.

8. (Original) A method as claimed in claim 7, wherein the value is chosen from a group that includes a set, an XML DOM node, and an XML DOM node list.

9. (Original) A method as claimed in claim 7, wherein the external data interface element is configured to have an entity reference attribute.

10. (Original) A method as claimed in claim 7, wherein the external data interface element is configured to have a return type attribute.

11. (Previously Presented) A method as claimed in claim 1, wherein establishing a set of computer-processable rules in accordance with a rules markup language includes creating a schema having an internal interface element and an external interface element.

12. (Original) A method as claimed in claim 1, further comprising creating a static document structure that can be resolved into one or more instances of a document that includes at least some content that is determined before and some content that is unchanged during and after a resolution process.

13. (Previously Presented) A method as claimed in claim 1, further comprising providing a data set configured to be processable by one or more rules built on the architecture for a set of computer-processable rules.

14. (Currently Amended) A method of generating a document, the method comprising:

creating a current transaction data set;

establishing at least one computer processable dynamic document structure;

establishing a set of computer-processable rules in accordance with a rules markup language, wherein the set of computer-processable rules are configured to be embedded in the at least one computer processable dynamic document[[s]] structure by creating a schema having a conditions element, a choose element, an iterators element, a functions element, and an external interface element that is configured to be resolved into a value;

configuring each rule in the set of computer-processable rules to be embedded in the at least one ~~or more~~ computer processable dynamic document structure[[s]] and to define content to be included in at least one instance of a document generated from the at least one ~~of the one or more~~ computer-processable dynamic document structure[[s]] when each rule is executed based on the current transaction data set;

creating a computer-implemented database;

storing each rule in the set of computer-processable rules in the database;

storing content in the database;

associating the stored content with one or more rules from the set of computer-processable rules in the database;

configuring the at least one ~~each~~ dynamic document structure to have a tree-architecture, to resolve to the at least one ~~or more~~ instance[[s]] of a document, and to include document content embedded with ~~that includes~~ one or more ~~embedded~~ rules from the set of computer-processable rules; and

resolving the at least one dynamic document structure, with a computer processor and in accordance with the current transaction data set, by executing the one or more ~~embedded~~ rules from the set of computer-processable rules embedded in the document content based on the current transaction data set to create a specific instance of a document in a static form.

15. (Currently Amended) A method of generating a document, the method comprising:

establishing a dynamic document structure;

creating a current transaction data set;

establishing a set of computer-processable rules in accordance with a rules markup language, wherein the set of computer-processable rules are configured to be embedded in documents, the set of computer-processable rules including a conditions element, a choose element, an iterators element, and a functions element and an external interface element, the set of computer-processable rules defining content to be included in the documents;

configuring each rule in the set of computer-processable rules to determine content to be included in at least one instance of a document generated based on a dynamic document structure when each rule is executed based on the current transaction data set;

creating a computer-implemented database;

storing each rule in the set of computer-processable rules in the database;

storing content in the database;

associating the stored content in the database with one or more rules from the set of computer-processable rules in the database;

configuring the ~~creating a~~ dynamic document structure to have a tree architecture and that can resolve to the at least one ~~or more~~ instance[[s]] of a document using the set of computer-processable rules and current transaction data set, the dynamic document structure including document content embedded with ~~that includes~~ one or more ~~embedded~~ rules based on the set of computer-processable rules;

creating a static document structure that can be resolved into one or more instances of a document that includes at least some content that is determined before and some content that is determined during a resolution process; and

resolving, with a computer processor and in accordance with the current transaction data set, the static document structure by executing the one or more rules from the set of computer-processable rules embedded in the document content based on the current transaction data set to create a specific instance of a document in a static form.

16. (Currently Amended) A method of assembling a document from a group of components, the method comprising:

creating a current transaction data set;

retrieving one or more cross-referenced document components from a data base based on the current transaction data set, the one or more document components configured to include document content and one or more embedded rules, the one or more embedded rules defining content to be included in documents when each embedded rule is executed based on the current transaction data set;

processing the one or more cross-referenced document components in a processor to generate a tree having a root node;

processing the tree beginning at the root node; ~~and~~

when one of the one or more embedded rules ~~a rule~~ is encountered, evaluating the rule based on the current transaction data set and replacing it with a value; and

creating a specific instance of a document in a static form based on the current transaction data set and the processing of the one or more cross-referenced document components.

17. (Previously Presented) A method as claimed in claim 16, further comprising establishing an architecture for a set of computer-processable rules.

18. (Previously Presented) A method as claimed in claim 17, wherein establishing an architecture for a set of computer-processable rules includes creating a schema having a conditions element.

19. (Previously Presented) A method as claimed in claim 17, wherein establishing an architecture for a set of computer-processable rules includes creating a schema having a choose element.

20. (Previously Presented) A method as claimed in claim 17, wherein establishing an

architecture for a set of computer-processable rules includes creating a schema having an iterators element.

21. (Previously Presented) A method as claimed in claim 17, wherein establishing an architecture for a set of computer-processable rules includes creating a schema having a functions element.

22. (Previously Presented) A method as claimed in claim 17, wherein establishing an architecture for a set of computer-processable rules includes creating a schema having a conditions element, a choose element, an iterators element, and a functions element.

23. (Previously Presented) A method as claimed in claim 17, wherein establishing an architecture for a set of computer-processable rules includes creating a schema having an external interface element that is configured to be resolved into a value.

24. (Original) A method as claimed in claim 23, wherein the value is chosen from a group that includes a set, an XML DOM node, and an XML DOM node list.

25. (Original) A method as claimed in claim 23, wherein the external data interface element is configured to have an entity reference attribute.

26. (Original) A method as claimed in claim 23, wherein the external data interface element is configured to have a return type attribute.

27. (Previously Presented) A method as claimed in claim 17, wherein establishing an architecture for a set of computer-processable rules includes creating a schema having an internal interface element and an external interface element.

28. (Currently Amended) A method of assembling a data structure from a group of components, the method comprising:

creating a current transaction data set;

retrieving one or more cross-referenced data structure components from a database based on the current transaction data set, the one or more data structure components configured to include document content and one or more embedded rules, the one or more embedded rules defining content to be included in documents when each embedded rule is executed based on the current transaction data set;

processing the one or more cross-referenced data structure components in a processor to generate a tree having a root node;

processing the tree beginning at the root node;

when one of the one or more embedded rules ~~a rule~~ is encountered, evaluating the rule based on the current transaction data set and replacing it with a value; and

creating a specific instance of a document in a static form based on the current transaction data set and the processing of the one or more cross-referenced document components.

29. (Previously Presented) A method as claimed in claim 28, further comprising establishing an architecture for a set of computer-processable rules.

30. (Original) A method as claimed in claim 28, further comprising establishing a list of data structures and performing each of the steps in claim 28 for each of the data structures.